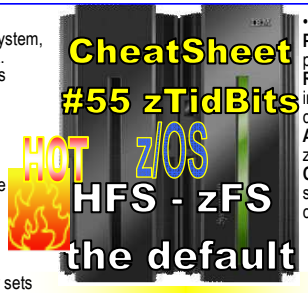
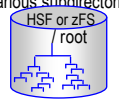


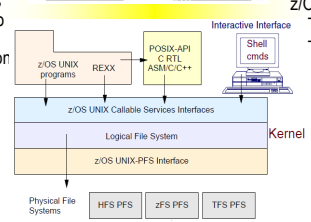
Begin with what these file systems are:

- A hierarchical file system (**HFS**) data set is an zOS data set that contains a POSIX-compliant hierarchical file system, which is a collection of files and directories organized in a hierarchical structure accessed using the z/OS UNIX.
- The file systems within HFS data sets have a tree structure based on a root directory with various subdirectories with files contained within directories.
  - The files within an HFS data set are identified by their path and file names.
  - HFS is used by UNIX System Services and its applications such as ftp, NFS, WebSphere Application Server for System z, and sundry subsystems, etc.
- HFS data sets were introduced in MVS/ESA with DFSMS/MVS Version 1Release 2.
- The support for HFS data sets was rewritten for DFSMS/MVS Version 1 Release 5 to increase the performance of an HFS.



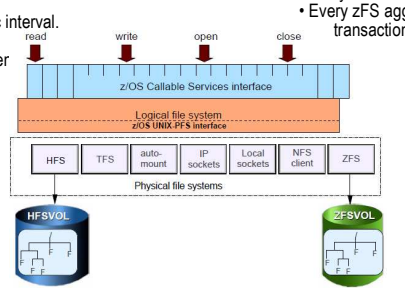
- The z/OS Distributed File Service (DFS) zSeries File System (**zFS**) is a z/OS UNIX file system that can be used in addition to the Hierarchical File System (HFS) and planned as the strategic **default file system** starting with zOS V1.11.

  NOTE: For hierarchical files that are switchable between zFS and HFS, the CustomPac Installation Dialog now sets the default to zFS. This is another step in the implementation of zFS as the strategic z/OS file system and applies to migration from z/OS V1R10 and z/OS V1R9. **This is not a required migration action, but highly recommended.**
  - zFS file systems contain files and directories that can be accessed with the z/OS hierarchical file system application programming interfaces on the z/OS operating system as follows: see diagram
    > An application interface composed of C interfaces, some of which are managed within the C Run-Time Library (RTL), while others access kernel interfaces to perform authorized system functions on behalf of the unauthorized caller
    > An interactive z/OS shell interface by shell users
    > The Physical File System (**PFS**) interface containg a set of protocols and calling interfaces between the logical file system (LFS) and the PFSs that are installed on z/OS UNIX as shown on right.
  - In a USS environment, UNIX programs and UNIX users access their files through these interfaces.
  - PFSs mount and unmount file systems and perform other file operations.
  - zFS provides significant performance gains in accessing files approaching 8K in size that are frequently accessed and updated where the access performance of smaller files is equivalent to that of HFS.
  - zFS provides reduced exposure to loss of updates by writing data blocks asynchronously, not waiting for a sync interval.
  - zFS is a logging file system and logs metadata updates.
    > If a system failure occurs, zFS replays the log when it comes back up to ensure that the file system is consister
  - zFS is a Physical File System (PFS) that is started by UNIX System Services (USS) during an IPL.
    > A physical file system is the part of the operating system that handles the actual storage and manipulation of data on a storage medium.
    > There are two types of PFSs, those that manage files and those that manage sockets: see diagram
      >> File management PFSs, such as HFS and zFS, deal with objects that have path names and that generally follow the semantics of POSIX files.
      >> Socket PFSs deal with objects that are created by the socket() and accept() functions and that follow socket semantics.



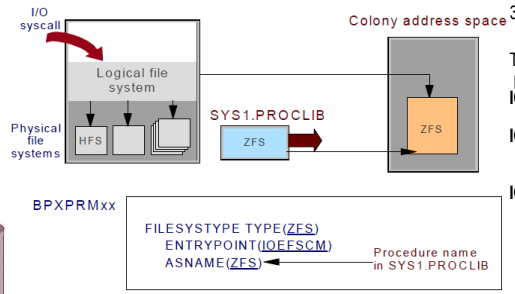
- zFS runs in a UNIX System Services (USS) colony address space which is an address space that is separate from the USS address space.

  NOTE: HFS runs inside the USS address space and zFS runs in its own address space, as shown here:



- A zFS aggregate [a] is a data set that contains zFS file systems. see diagram
  - The aggregate is a VSAM Linear Data Set (VSAM LDS) and is a container that can contain one or more zFS file systems.
  - An aggregate can only have one VSAM LDS, but it can contain an unlimited number of file systems.

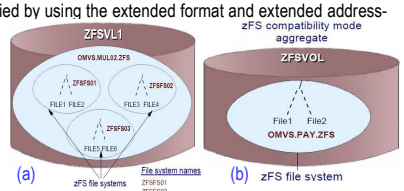
  NOTE: A compatibility mode aggregate [b] can contain only one zFS file system, making this type of aggregate more like an HFS file system.
  - The name of the aggregate is the same as the VSAM LDS name.
  - Sufficient space must be available on the volume or volumes, as multiple volumes may be specified on the DEFINE of the VSAM LDS (see below).
  - VSAM LDSs greater than 4 GB may be specified by using the extended format and extended address-ability in the SMS data class of the data set.




```
//ZFSJOB   JOB ,'ZFS NewAggr',
//         CLASS=A,MSGCLASS=X,MSGLEVEL=(1,1),TIME=1440
//DEFINE   EXEC   PGM=IDCAMS
//SYSPRINT DD     SYSOUT=*
//SYSUDUMP DD     SYSOUT=*
//AMSDUMP  DD     SYSOUT=*
//DASD0    DD     DISP=OLD,UNIT=3390,VOL=SER=TOTZF1
//SYSIN    DD     *
  DEFINE CLUSTER (NAME(OMVS.MUL01.ZFS) VOLUMES(TOTZF1) -
   LINEAR MEGABYTES(20 10) SHAREOPTIONS(3))
...
//*
```

sample DEFINE

Note the ending qualifier



- The zFS file system has a cache for file system metadata, which includes directory contents and the data of files that are smaller than the aggregate block size.
  - The setting of this cache size is *important to performance* because zFS references the file system metadata frequently since synchronous reads of metadata increase I/O rates to disk and server response times.
  - Metadata consists of things like owner, permission bit settings, and data block pointers.
  - The metadata cache is stored in the zFS primary address space; its default size is 32 MB.
  - Since the metadata cache only contains metadata and small files, it normally does not need to be nearly as large as the user file cache.

⚠ Beginning in z/OS V1R11, you can no longer attach zFS multi-file system aggregates that are shared across systems in a sysplex. IBM has previously recommended that multi-file system aggregates not be shared in a sysplex environment. Attaching zFS compatibility mode aggregates, which have a single file system per data set, will continue to be supported in all environments.

- zFS provides many features and benefits:

Performance- zFS provides significant performance gains in many customer environments. zFS provides additional performance improvements when running sysplex-aware in a shared file system environment.

Restart- zFS reduces the exposure to loss of updates. zFS writes data blocks asynchronously and does not wait for a sync interval. zFS is a logging file system. It logs metadata updates. If a system failure occurs, zFS replays the log when it comes back up to ensure that the file system is consistent.

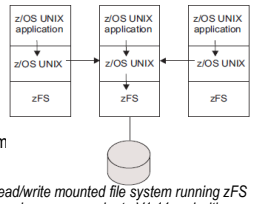
Aggregate movement- As a part of supporting read-write mounted file systems that are accessed in sysplex-aware mode, zFS automatically moves zFS ownership of a zFS file system to the system that has the most read-write activity.

Cloning- As an optional function, zFS allows the administrator to make a read-only clone of a file system in the same data set. This clone file system can be made available to users to provide a read-only point-in-time copy of a file system. The clone operation happens relatively quickly and doesn't take up much additional space since only the metadata is copied.

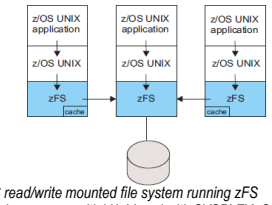
- zFS has its own concept of **file system ownership** known as the ***zFS file system owner*** **.**
- This is also typically the system where the file system is first mounted in a sysplex-aware environment.
- File requests to sysplex-aware file systems are sent directly to the local zFS PFS rather than being forwarded to the z/OS UNIX file system owner.

  This concept is shown here. →

  The local zFS PFS forwards the request to the zFS file system owner, if necessary. The z/OS UNIX file system owner can be different from the zFS file system owner.



zFS read/write mounted file system running zFS non-sysplex aware – prior to V1.11 and with SYSPLEX=OFF



zFS read/write mounted file system running zFS sysplex aware – with V1.11 and with SYSPLEX=ON

- zFS file systems can be shared in a sysplex environment meaning that users in a sysplex can access a zFS file in a file system that is owned and mounted on another system in the sysplex.
- Every zFS aggregate contains a log file that is created when the aggregate is formatted and is used to record transactions describing changes to the file system structure.
- zFS provides a recovery mechanism that uses a zFS file system log to verify or correct the structure of an aggregate using a recovery mechanism which can also be invoked by a utility program named **IOEAGSLV**.
- When you perform a system restart, a recovery program known as the *salvager* uses the zFS file system log to return consistency to a file system by running recovery on the aggregate on which the file system resides.
- In z/OS V1R10 and above, you can use the **man** command view the descriptions of zFS command manual pages.
- zFS provides utility programs and z/OS UNIX commands (**zfsadm**) to assist in the customization of the aggregates and file systems to be used by system administrators.

  The **zfsadm** command can be run as a UNIX shell command from:
  1. A z/OS UNIX system services shell (OMVS) or a (z/OS UNIX) telnet session
  2. A batch job using the BPXBATCH utility program
  3. TSO foreground or in batch mode using z/OS UNIX APIs (SYSCALL commands, Callable Services)

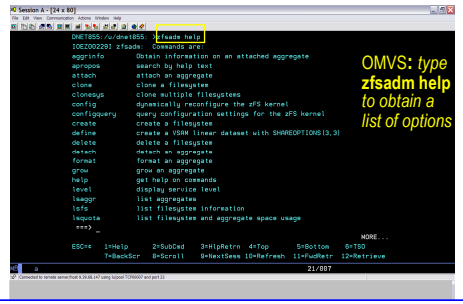
The following **utility programs** are provided:

IOEAGFMT A utility program to format an aggregate.

IOEAGSLV A utility program to scan an aggregate and report inconsistencies.

IOEZADM A utility program that allows **zfsadm** commands to be issued using JCL. It is also supported to be run in a TSO/E environment.

- zFS ***performance*** is dependent on many factors and zFS provides performance information to help the administrator determine bottlenecks.
- The IOEFSPRM file contains many tuning options that can be adjusted.
  - IOEPRMxx contains the configuration options that control the zFS aggregates to be attached at the start-up.
  - The IOEPRMxx files are contained in the logical parmlib concatenation, defined on the PARMLIB parameter in SYS1.PARMLIB(LOADxx) member.
- The output of the system **MODIFY ZFS, QUERY** commands provide feedback about the operation of zFS. →



Command/subcommand	Command description
zfsadm aggrinfo	Obtain information on attached aggregate
zfsadm apropos	Display first line of help entry
zfsadm attach	Attach an aggregate
zfsadm clone	Clone a file system
zfsadm clonesys	Clone multiple file systems
zfsadm create	Create a file system
zfsadm define	Define a VSAM linear data set
zfsadm delete	Delete a file system
zfsadm detach	Detach an aggregate
zfsadm format	Format a VSAM LDS as an aggregate
zfsadm grow	Grow an aggregate
zfsadm help	Get help on commands
zfsadm lsaggr	List all currently attached aggregates
zfsadm lsfs	List all file systems on a aggregate or all
zfsadm lsquota	Show quotas for file systems & aggregates
zfsadm lssys	Shows the members in a sysplex
zfsadm quiesce	Quiesce an aggregate and all file systems
zfsadm rename	Rename a file system
zfsadm setquota	Set the quota for a file system
zfsadm unquiesce	Make aggregates & file systems available
zfsadm config	Modify current configuration options
zfsadm configquery	Display current configuration options
zfsadm query	Query or reset the performance counters